Feature Structure Matching for Multi-source Sentiment Analysis with Efficient Adaptive Tuning

Rui Li, Cheng Liu*, Yu Tong, Jiang Dazhi*
Department of Computer Science, Shantou University
{ruili, cliu, tongyu, dzjiang}@stu.edu.cn

Abstract

Recently, fine-tuning the large pre-trained language models on the labeled sentiment dataset achieves appealing performance. However, the obtained model may not generalize well to the other domains due to the domain shift, and it is expensive to update the entire parameters within the large models. Although some existing domain matching methods are proposed to alleviate the above issues, there are multiple relevant source domains in practice which makes the whole training more costly and complicated. To this end, we focus on the efficient unsupervised multi-source sentiment adaptation task which is more challenging and beneficial for real-world applications. Specifically, we propose to extract multi-layer features from the large pre-trained model, and design a dynamic parameters fusion module to exploit these features for both efficient and adaptive tuning. Furthermore, we propose a novel feature structure matching constraint, which enforces similar feature-wise correlations across different domains. Compared with the traditional domain matching methods which tend to pull all feature instances close, we show that the proposed feature structure matching is more robust and generalizable in the multi-source scenario. Extensive experiments on several multi-source sentiment analysis benchmarks demonstrate the effectiveness and superiority of our proposed framework.

Keywords: Pre-trained language model, Multi-Source Sentiment analysis, Efficient domain adaptation, Features structure matching

1. Introduction

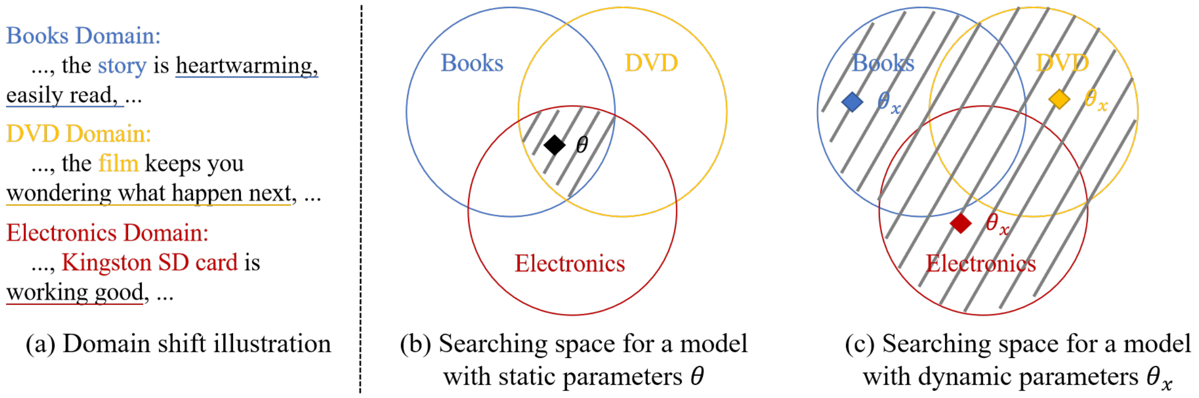
Sentiment analysis (SA) (Cambria et al., 2020) is an important task in the NLP field, which aims to predict the sentiment label (i.e., positive or negative) with a given sentence (Susanto et al., 2022) and has wide applications, *e.g.,* conversation sentiment recognition (Tu et al., 2022), public opinion monitoring (Lin and Luo, 2020). Previous methods adopt relatively small networks to make predictions, *i.e.*, Convolutional Neural Networks (CNN) or Long Short Term Memory Networks (LSTM) (Rhanoui et al., 2019). While, with the advent of Transformer (Vaswani et al., 2017), various large pre-trained language models (Devlin et al., 2019; Yang et al., 2019b) significantly improve the performance on the SA task, which often include two stages. First, pre-training the transformer-based model on the large-scale raw texts with some specific self-supervision tasks. Second, fine-tuning the pre-trained model on the newly-collected labeled sentiment dataset. Despite their great progress, there still exist two issues:

(1) The fine-tuned model may not generalize well under different distributions (Wilson and Cook, 2020), since text from different domain contains different subjects or sentiment descriptions, as illustrated in Figure 1 (left).

(2) The pre-trained language models often have large parameters, fine-tuning the entire set of parameters is time-consuming and requires large GPU memory costs (Sung et al., 2022).

For the first issue, a straightforward way is to adopt domain adaptation techniques (Wilson and Cook, 2020). Most existing methods focus on the single-source domain adaptation problem, and aim to obtain domain-invariant features by pulling the feature instances close (Du et al., 2020). While, there often exist multiple source domains in real practice (Guo et al., 2018), that can be leveraged to improve the performance. However, due to various multi-source domain distributions, pulling features from all the domains together with conventional methods may compromise the generalization of the adapted model (Zhou et al., 2021), since the shared (domain-invariant) information is significantly reduced (Figure 1). Besides, Gulrajani and Lopez-Paz (2021) also demonstrate most strict domain matching methods hurt the model's generalization and lead to degraded performance in multi-source domain scenarios, since forcing all features close can distort their semantic information. On the other hand, we assume that features across different domains should follow a similar feature-wise structure for fine-grained distribution alignments instead of pulling them arbitrarily. Therefore, we propose a Features Structure Matching (FSM) constraint, which is shown to be robust and generalizable. Specifically, we exploit the feature structure by computing multi-order feature-wise correlation matrices, and enforce these matrices to be consistent across different domains. We empirically demonstrate that FSM constraint achieves superior

*Corresponding authors


Books Domain:
..., the story is heartwarming, easily read, ...
DVD Domain:
..., the film keeps you wondering what happen next, ...
Electronics Domain:
..., Kingston SD card is working good, ...
Books
DVD
Electronics
θ
Books
DVD
Electronics
θ_x
θ_x
θ_x

(a) Domain shift illustration

(b) Searching space for a model with static parameters θ

(c) Searching space for a model with dynamic parameters θ_x

Figure 1: Illustration the problem of the multi-source domain shift, and the comparison of the model with static parameters θ and dynamic parameters θ_x, *i.e.*, conditioned on input x.

results than previous methods on multiple multi-source sentiment benchmarks.

For the second issue, some previous methods insert small trainable blocks into the large model and only update their parameters during training (Houlsby et al., 2019; Li and Liang, 2021; Liu et al., 2022). However, these methods still involve the large backbone during adaptation training, and the gradients will back-propagate through the entire backbone for computing the corresponding gradients of the inserted parameters (Sung et al., 2022). To this end, we propose a new adaptive tuning strategy, which does not involve the backbone during training. Thus, the adaptation training latency can significantly reduced. Specifically, we first extract the features from multi-layers of the large language models as input. Then, for more elastic adaptability, we further design a Dynamic Parameters Fusion (DPF) module which can adjust the network parameters according to each input, so that the dynamic model can adaptively fit to various input (Li et al., 2021), as shown in Figure 1 (right). In this case, multiple diverse source domains can be leveraged to enhance the generalization of the model.

As discussed above, there is a strong motivation to develop an efficient and generalizable multi-source sentiment analysis framework. We propose two corresponding modules, which are FSM and DPF for both transferability and efficiency. To summarize, the contributions are as follows:

- We propose a Feature Structure Matching (FSM) constraint, which focuses on matching fine-grained feature-wise correlations and leads to superior results in the multi-source setting.
- A novel Dynamic Parameters Fusion (DPF) module is designed for multi-layer feature processing, and the corresponding large backbone is not involved for efficient adaptation.
- Extensive experimental results demonstrate that our proposed framework achieves state-of-the-art performance on multiple multi-source sentiment adaptation benchmarks.

Section 2 introduces the related work. Section 3 gives the details of the proposed framework, followed by the experimental results in Section 4. Finally, section 5 draws the conclusion.

2. Related Work

In this section, we introduce some representative works about sentiment analysis, domain adaptation, and dynamic networks.

Sentiment analysis is one of the important tasks in the NLP field (Wankhade et al., 2022). Previous methods use word2vec or GloVe embedding as the text features, which can not capture the contextual information and results in sub-optimal performance. Recently, fine-tuning the large language models achieves significant improvement on the sentiment analysis task (Devlin et al., 2019; Yang et al., 2019b), which can be attributed to the large model capacity and the self-attention module (Vaswani et al., 2017) for capturing the contextual information. With the development of large language models, they are dominating various tasks in the NLP field. However, fine-tuning large language models is still cumbersome, and the obtained model can not generalize well to the other domains due to the domain shift (Du et al., 2020).

Domain adaptation receives much attention in the deep learning field (Zhao et al., 2022), which aims to transfer knowledge from the labeled source domain to the unlabeled target domain. The mainstream is to learn the domain-invariant feature by minimizing a specific distribution distance across domains, such as Maximum

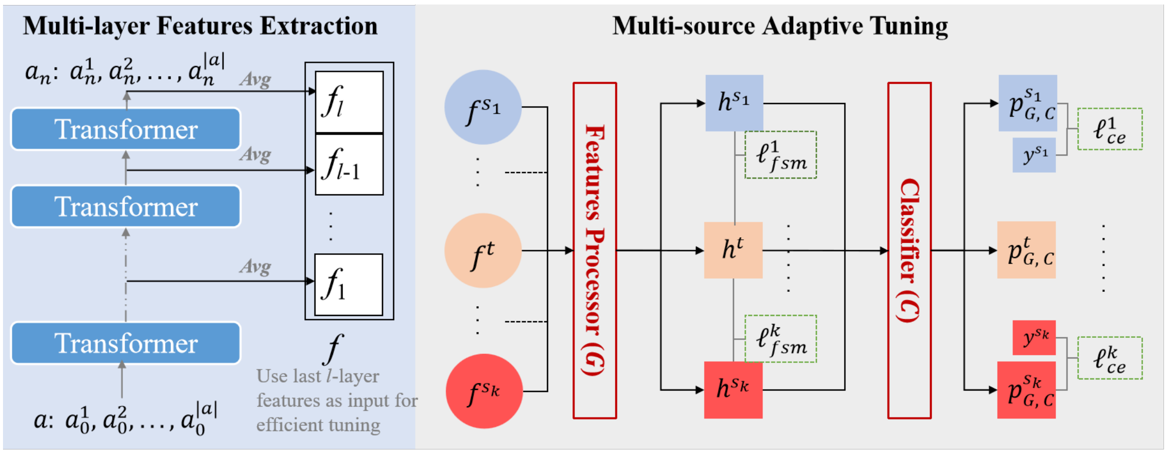


Figure 2: Illustration of our overall framework, which includes multi-layer features extraction and multi-source adaptive tuning. It is noted that we use the extracted features as input during adaptive tuning, which implies that we only need to update the trainable parameters within G and C denoted with red blocks during adaptation.

Mean Discrepancy (MMD) (Long et al., 2019), covariance distance (Sun and Saenko, 2016), *etc.* On the other hand, Du et al. (2020) introduce a discriminator to pull features close via adversarial training (Ganin et al., 2016; Goodfellow et al., 2014). Some previous works also leverage pivot words (Ziser and Reichart, 2018) to improve the sentiment analysis. Most works focus on the single-source domain adaptation, while the training data often include multiple domains in real-world applications. Some relevant methods naturally adopt single-source adaptation techniques. For example, mixture-of-experts (Guo et al., 2018) simply extends the MMD-based method by aligning every domain pair. Multi-source Domain Adversarial Networks (MDAN) (Zhao et al., 2018) extends DANN (Ganin et al., 2016) with multiple domain classifiers. Hoffman et al. (2018) proposes a theory that determines the distribution-weighted combination solution for the multi-source adaptation problem. Therefore, an improved strategy is to assign different weights for each source domain based on the distribution discrepancy to the target domain, and the final prediction is a weight combination of the outputs from corresponding source classifiers (Dai et al., 2020; Fu and Liu, 2022; Hoffman et al., 2018). However, some recent works observed that arbitrarily pulling feature instances close may hurt features' semantic information and sacrifice the model's generalization (Gulrajani and Lopez-Paz, 2021). In contrast to the traditional domain matching methods, we propose a feature structure matching constraint which focuses on feature-wise correlation similarities. The original distribution is less distorted, and the overall adaptation training is more robust. Therefore, different domain distributions can be carefully aligned for better performance.

Dynamic networks are designed to adjust the model's architecture or parameters conditioned on inputs, which can increase the model's capacity and adaptability (Han et al., 2022; Xu and McAuley, 2023). CondConv (Yang et al., 2019a) and DY-CNNs (Chen et al., 2020b) select the optimal combination of the convolution parameters, which increase the model capacity with marginal cost. Han et al. (2022) reports several strategies of dynamical computation in the NLP field. Li et al. (2021) and Li et al. (2022) demonstrate that dynamic networks can achieve improved results on the multi-source domain adaptation tasks. Inspired by these works, we propose a dynamic parameters fusion module, which is conditioned on the global features and outputs parameters for the dynamic fully-connected layers. Therefore, the dynamic layer can adjust suitable parameters for more adaptive tuning.

3. Method

In this section, we introduce the details of our method for multi-source sentiment adaptation. We first present the task formulation and the motivation. Then, we explain the architectures and the loss functions used in the framework. Last, we give the overall training procedure of our method.

3.1. Formulation and Overall Framework

We focus on the unsupervised multi-source sentiment analysis task, where there are K labeled source domains $\mathcal{S} = \{\mathcal{S}_k\}_{k=1}^K$, *i.e.*, $\mathcal{S}_k = \{x_m^{\mathcal{S}_k}, y_m^{\mathcal{S}_k}\}_{m=1}^{|\mathcal{S}_k|}$ and an unlabeled target domain $\mathcal{T} = \{x_m^{\mathcal{T}}\}_{m=1}^{|\mathcal{T}|}$, $|\cdot|$ indicates number of instances in the domain. Not only each source domain has different distributions with the target domain (*i.e.*,

$\mathcal{P}_{\mathcal{S}_k} \neq \mathcal{P}_{\mathcal{T}}$), but also every two source domains have different distributions (*i.e.*, $\mathcal{P}_{\mathcal{S}_k} \neq \mathcal{P}_{\mathcal{S}_q}$). Therefore, it is more complicated than the single-source domain adaptation problem, and our goal is to train a sentiment analysis model on the given data, which can generalize well on the target domain.

Figure 2 shows the overall framework of our method. We adopt a pre-trained language model as the backbone due to their superior performance. For efficient adaptation, we extract last-l layer features and concatenate them as the input for the subsequent multi-source adaptive tuning. It is noted that the adaptation training stage is independent from the backbone, the trainable model only consists of a Features Processor (G) for processing the multi-source multi-layer features, and a Classifier (C) for the sentiment prediction. To obtain a more generalizable model, we design a dynamic parameters fusion module within G. Each module and objective will be detailed as follows.

3.2. Multi-layer Features Extraction

As shown in Figure 2 (left), our framework adopts a pre-trained language model as the backbone for the feature extraction. A sentence is denoted as $a = [{a_0}^1, {a_0}^2, ..., {a_0}^{|a|}]$ (assume the raw text in the 0^{th} layer), $|a|$ is the number of words in a. The backbone often consists of multiple transformer layers, and the corresponding output for l^{th}-layer layer denotes $a_l = \texttt{Transformer}_l(a_{l-1}) = [{a_l}^1, {a_l}^2, ..., {a_l}^{|a|}]$, ${a_l}^i$ is the i^{th} word embeddings in the l^{th}-layer layer, and so on so forth. In order to save memory and computation costs, we tend to use the extracted features from different transformer layers as input for the subsequent adaptation, and leave the backbone freezing. In addition, sentences could have different word lengths, and to maintain the same input dimensions, we average all the word embeddings within a_l as the extracted feature of the l^{th}-layer, *i.e.*, $f_l = \texttt{Avg}(a_l)$. Therefore, the corresponding sentence features from multiple transformer layers are $f = \{f_1, f_2, ..., f_l\}$, where we use last l-layer features in the experiments.

3.3. Dynamic Parameters Fusion

Sun et al. (2019) demonstrates the fine-tuning different layers of BERT (Devlin et al., 2019) could have different performance, which indicates that different layer features contain different aspects information of the sentence. On one hand, it is important to fully leverage the multi-layer features for better results; On the other hand, the adaptation model should be more elastic to fit to various domains. Therefore, we introduce a novel Dynamic Parameters Fusion (DPF) module within G, which can produce the parameters of a specific layer conditioned on the global input feature. This indicates that our framework can automatically adjust the model's parameters for various input from different domains. In this case, the model is more generalizable and can be easily adapted to the target domain.

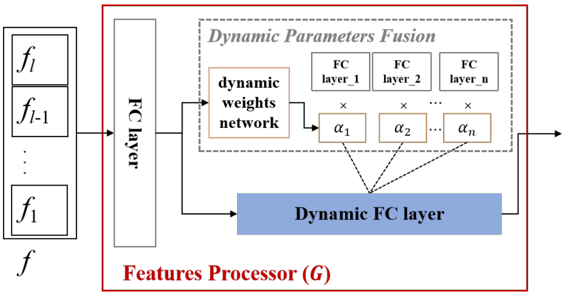


Figure 3: The architecture of G which includes a dynamic parameters fusion module. The activation function is omitted for simplicity.

Specifically, we define N fully-connected layers $\{\mathbf{W}_i, \mathbf{b}_i\}_{i=1}^N$ within the DPF, $\mathbf{W}$ and $\mathbf{b}$ denote the parameters of the weight and bias. Instead of computing a weight for each layer feature for fusion, we first employ a fully-connected layer ($\texttt{FC}$) to extract global information for DPF, so that DPF can leverage all the information within the multi-layer features. The output of DPF is the dynamic parameters ($\mathbf{W_d}, \mathbf{b_d}$) for a dynamic fully-connected layer. The dynamic fully-connected layer can be viewed as aggregating N fully-connected layers with different coefficients α, which is defined as follows:

$$\mathbf{W_d} = \sum_{i=1}^{N} \alpha_i(\texttt{FC}(f))\mathbf{W}_i \quad \mathbf{b_d} = \sum_{i=1}^{N} \alpha_i(\texttt{FC}(f))\mathbf{b}_i$$

$$\text{s.t.} \quad \alpha_i(\texttt{FC}(f)) \in [0,1], \quad \sum_{i=1}^{N} \alpha_i(\texttt{FC}(f)) = 1 \qquad (1)$$

where $\alpha_i(\cdot)$ denotes the coefficient for the i^{th} fully-connected layer, which is conditioned on $\texttt{FC}(f)$. As shown in Figure 2 (right), we use a small dynamic weights network that includes two fully-connected layers with N neuron outputs to derive the coefficients $\{\alpha_i\}_{i=1}^N$ in our experiments (Chen et al., 2020b). Based on the above descriptions, the dynamic fully connected layer can adjust its parameters based on various inputs, so that the adapted model can fully exploit the multi-layer features and is allowed to fit multiple domains.

3.4. Feature Structure Matching

To further improve the performance on the unlabeled target domain, we need to match the domain distributions. However, pulling the features arbitrarily may distort the original semantic information. Therefore, we propose to match feature structures, which correspond the high-order correlations (Chen et al., 2020a) among different dimensions within the features. In this case, feature distributions can be adjusted in a fine-grained manner. We demonstrate that our feature structure matching constraint is more robust and suitable for multi-source domain adaptation.

We use the multi-order feature correlation matrices S_h to define the structures of feature h, which is defined as follows:

$$\begin{aligned} S_h &= \{h^{\otimes p}\}_{p=2}^{\infty} \\ h^{\otimes p} &= \underbrace{h \otimes h \otimes ... \otimes h}_{p} \in \mathbb{R}^{c^p} \end{aligned} \tag{2}$$

where $\otimes$ denotes the outer product operation, and c is the number of dimensions in the h. S_h includes multiple correlation matrices from second-order to infinity-order. Each correlation matrix is only related to the number of dimensions c.

In addition, when $p = 2$, the corresponding second-order correlation matrix is $h^{\otimes 2} = h \otimes h = (h^T \times h) \in \mathbb{R}^{c \times c}$, which is exactly the Gram matrix (Johnson et al., 2016) and is widely used in the image style transfer. Therefore, the feature correlation matrices contain the distribution information. As shown in Figure 2 (right), the feature structure matching between the source domain and the target domain can be expressed as follows:

$$\ell_{fsm} = \sum_{p=2}^{\infty} \frac{1}{c^p} || \frac{1}{b} \sum_{m=1}^{b} {h_m^s}^{\otimes p} - \frac{1}{b} \sum_{m=1}^{b} {h_m^t}^{\otimes p} ||_F^2 \tag{3}$$

where b denotes the batch size during training. h_m denotes a processed feature by G, i.e, $h_m = G(f_m)$. Footnote s and t denote the source domain and the target domain, respectively. Therefore, ℓ_{fsm} tends to match all the high-order feature-wise correlations. Noted that we remove the first-order matching, which is equivalent to the linear MMD constraint. We observe that first-order matching is too strict, which may hurt the model performance under multi-source scenarios.

3.5. Overall Training Procedures

We define the final prediction model as the composition of G and C, and the classification output as $p_{G,C}(f)$. Therefore, the overall objectives can be expressed as follows:

$$\min_{\theta_G, \theta_C} \frac{1}{K} \sum_{k=1}^{K} [\ell_{ce}^k(G, C) + \lambda_d \ell_{fsm}^k(G)] \tag{4}$$

where $\ell_{ce}^k(G, C) = -\frac{1}{b} \sum_{m=1}^{b} y_m{}^{S_k} \log p_{G,C}(f_m{}^{S_k})$ is the supervised cross entropy in the k^{th} source domain, $\ell_{fsm}^k(G)$ is the feature structure matching loss between the k^{th} source domain and the target domain. λ_d is a hyperparameter that trade-offs their effects.

We proceed the training by optimizing G and C based on the averaged losses over all the source domains as shown in Eq. 4. The detailed optimization procedure is summarized in Algorithm 1. During the test, we adopt the composition of $G \circ C$ as the final model.

Algorithm 1 Pseudo-code of our efficient multi-source sentiment analysis model

```
Input: Extracted last $l$ pre-trained language model features for all the domains (including $K$ source domains and the target domain), learning rates $\zeta$ for the features processor $G$ and the classifier $C$;
Output: $\theta_G, \theta_C$;
for $step$ = $1$ to $all\_steps$ do
    for each mini-batch $b$ do
        for $k$ = $1$ to $K$ do
            Compute the source supervised cross-entropy loss: $\ell_{ce}^k$;
            Compute the features structure matching loss: $\ell_{fsm}^k$;
        end for
        averaging the losses over all the domains based on Eq. 4;
        Update $G$ and $C$ via:
            $\theta_C, \theta_G \leftarrow \texttt{Adam}(\nabla_{\theta_G,\theta_C}(\frac{1}{K}\sum_{k=1}^{K}[\ell_{ce}^k(G,C) + \lambda_d \ell_{fsm}^k(G)]), \theta_G, \theta_C, \zeta)$;
    end for
end for
```

4. Experiments

In this section, we will evaluate our framework on two widely used sentiment analysis benchmarks. We first introduce the experimental settings, which include dataset descriptions and implementation details. Then, we compare our method with the recent state-of-the-art multi-source domain adaptation methods. Finally, extensive ablation studies and modal analysis are presented to verify the effectiveness of our framework.

4.1. Experimental Settings

Amazon-reviews dataset [1]: contains reviews from four-product domains, namely, Books, DVD, Electronics, Kitchen. Each domain includes 1,000 positive and negative reviews, respectively. Following a similar adaptation protocol to Li et al. (2022), we

[1] https://www.cs.jhu.edu/ mdredze/datasets/sentiment/

Method	D, E, K → B	B, E, K → D	B, D, K → E	B, D, E → K	Avg.
Previous methods					
DANN (Ganin et al., 2016)	0.779	0.789	0.849	0.864	0.820
MDAN (Zhao et al., 2018)	0.786	0.807	0.853	0.863	0.827
MoE (Guo et al., 2018)	0.794	0.834	0.866	0.880	0.843
2ST-UDA (Dai et al., 2020)	0.799	0.839	0.851	0.877	0.841
CTDA (Fu and Liu, 2022)	0.800	0.839	0.866	0.880	0.846
AML (Li et al., 2022)	0.852	0.856	0.880	0.892	0.870
Single-best	0.861	0.858	0.883	0.879	0.870
Source-combined	0.854	0.863	0.887	0.892	0.874
Our model	**0.872**	**0.867**	**0.895**	**0.900**	**0.884**

Table 1: Comparison of multi-source unsupervised sentiment adaptation on Amazon-reviews datasets. The best results are denoted with **bold**.

Method	B, D, E, K → AL	B, D, E, K → AP	Avg.
AML (Li et al., 2022)	0.850	0.695	0.772
Single-best	0.860	0.698	0.764
Source-combined	0.863	0.695	0.756
Our model	**0.876**	**0.703**	**0.789**

Table 2: Adaptation performance from multiple product review domains (Amazon) to one of air-travel review domains (Skytrax). The best results are denoted with **bold**.

conduct four unsupervised multi-source sentiment adaptation tasks by treating any one as the target domain and the remaining domains as the source domains.

Skytrax-reviews dataset [2]: includes two air-travel related reviews from *skytrax* website, *i.e.,* Airline (AL) and Airport (AP). To align with Amazon view datasets, we randomly sample 1,000 positive and 1,000 negative reviews from AL and AP domains for training. Since they are very different from the product domains, the domain discrepancy between the Amazon views and Skytrax reviews is large. We use all four product datasets as source domains and one of Skytrax view datasets as the target domain. These two multi-source sentiment adaptation tasks can verify the effectiveness of our method under challenging settings.

Implementation details: In all experiments, the pre-trained BERT_{base}-uncased (Devlin et al., 2019) is adopted to extract features. For fair comparison and efficiency as reported in Merchant et al. (2020), we use features from the last 4 transformer layers. The computation costs for higher-order correlation matrices is surging and the improvement is limited when $p \geq 4$. Therefore, we adopt second- and third-order correlation matrices to approximate the feature structures. We adopt a similar experimental setting with the recent work (Li et al., 2022) for fair comparison. All the datasets are public and split into the training and test set. We use Adam (Kingma and Ba, 2015) optimizer and set λ_d to 10^2, the learning rate to 5×10^{-5} in all the experiments.

4.2. Experimental Results

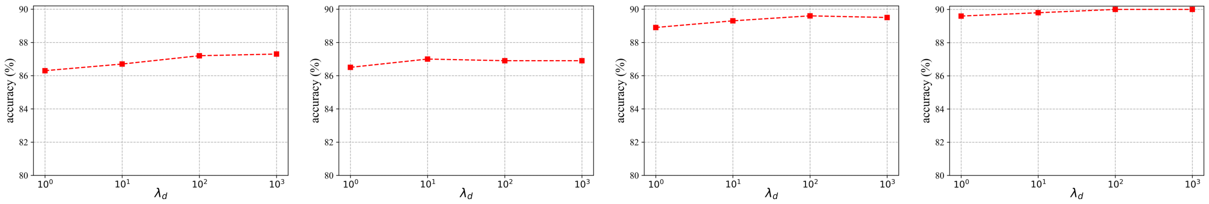
Results on Amazon-reviews benchmarks: Table 1 reports the accuracy of our method and recent multi-source unsupervised adaptation methods on Amazon-reviews benchmarks. It is obvious that our method achieves the best performance on all four multi-source sentiment adaptation tasks. Note that most previous methods adopt word embeddings, which are less informative. While, we use the BERT features as input, thus, better performance can be expected. For example, the average performance of CTDA (Fu and Liu, 2022) is 84.6%, our model can significantly outperform it by around 4 percentage points. In addition, training with pseudo labels achieves impressive performance on the various domain adaptation tasks (Liu et al., 2021), recently. AML (Li et al., 2022) also adopts BERT backbone for training, and involves pseudo-label training with multiple classifier heads. Our model is based on distribution matching and achieves average accuracy of 88.4%, which surpass AML by 1.4 percentage points. We also demonstrate that our model is orthogonal with the recent self-training techniques for more enhanced performance (shown in Sect. 4.4).

Adaptation Results from Amazon to Skytrax: Table 2 compares the performance of sentiment adaptation from Amazon product reviews to airline (AL) and airport (AP) reviews, respectively. Our model still outperforms the previous pseudo-label

[2] https://github.com/quankiquanki/skytrax-reviews-dataset

Method	D, E, K → B	B, E, K → D	B, D, K → E	B, D, E → K	Avg.
MMD (1^{st}-order matching)	0.859	0.860	0.885	0.892	0.874
ℓ_{fsm} w/o. 2^{nd}-order matching	0.864	0.864	0.890	0.893	0.878
ℓ_{fsm} w/o. 3^{rd}-order matching	0.868	0.866	0.893	0.896	0.881
ℓ_{fsm}	**0.872**	**0.867**	**0.895**	**0.900**	**0.884**

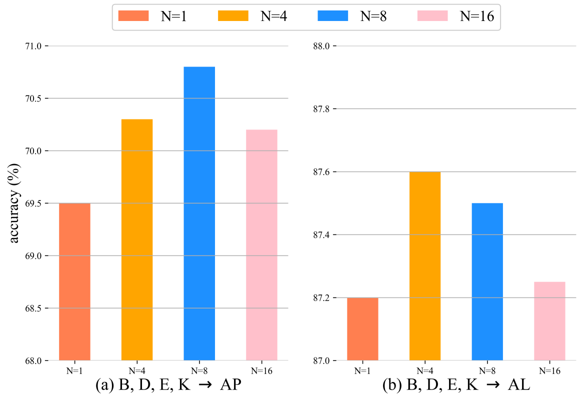
Table 3: Ablation study of the effects of each order correlation constraint in the proposed ℓ_{fsm}. The best results are denoted with **bold**.



(a) D, E, K → B (b) B, E, K → D (c) B, D, K → E (d) B, D, E → K

Figure 4: Hyperparameter analysis of λ_d on the Amazon view dataset.

training method (AML) by around 2 percentage points on average. '*Single-best*' indicates the best performance is achieved with a single source domain, which often follows more similar distributions to the target domain. '*Source-combined*' indicates the performance is achieved by training on the combined source domains, which is regarded as a strong baseline. The performance of '*Source-combined*' is sometimes worse than that of '*Single-best*', even with more data. We speculate that various source domain distributions cause conflicts, and simply combining all the source datasets can hurt the model's performance. On the contrary, our model can consistently outperform both baselines, which verifies the effectiveness of our method.



(a) B, D, E, K → AP (b) B, D, E, K → AL

Figure 5: The effect of number of Fully-Connected layers in DPF.

4.3. Ablation Study

Effectiveness of FSM: We first validate the effectiveness of the Feature Structure Matching constraint by removing the corresponding loss ℓ_{fsm} in the Eq. 4. As shown in Table 1-2, our model improves the baselines in all the multi-source adaptation tasks, which verifies that FSM can alleviate the domain shift for better performance. In addition, we study the effects of different order correlation constraints within ℓ_{fsm}. As shown in Table 3, removing any-order correlation constraints decreases the adaptation performance. We also try to add higher-order correlation constraint, *i.e.*, $p >= 4$, which only brings about marginal performance improvement, but incurs significant computation cost. In addition, our ℓ_{fsm} is also superior to the first-order statistics matching constraint, *i.e.,* MMD $(p = 1)$ with the linear kernel or Gaussian kernels. MMD is observed to be sensitive to λ_d, and often fails to converge in our experiments. This demonstrates the superiority of FSM in multi-source settings. We speculate that MMD constraint tends to arbitrarily pull the feature instances from all domains close, which can hurt their semantic information. While our ℓ_{fsm} focuses on aligning feature structures, the semantic information of feature is less affected.

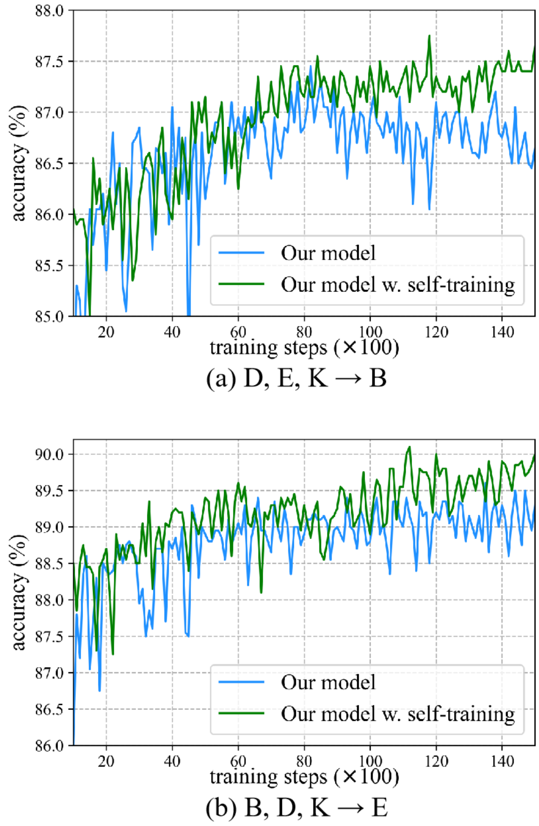
Effectiveness of DPF: We further remove the Dynamic Parameters Fusion module within G to verify its effectiveness, the corresponding module becomes a static network which is equivalent to a fully-connected layer. The comparison performances are shown in Table 4. It is obvious that our proposed DPF module can consistently improve the overall performance, which validates that adjusting the model's parameters based on different inputs can enhance its generalization since the

Method	B, D, E, K → AL	B, D, E, K → AP	Avg.
w/o. DPF (static model)	0.872	0.695	0.783
w/. DPF (dynamic model)	**0.876**	**0.703**	**0.789**

Table 4: Ablation study the effects of the proposed DPF module. The best results are denoted with **bold**.

Method	D, E, K → B	B, E, K → D	B, D, K → E	B, D, E → K	Avg.
w/. BERT	0.872	0.867	0.895	0.900	0.884
w/. Sentence-BERT	0.896	0.893	0.920	0.931	0.910

Table 5: Comparison of the adaptation results with the backbone of BERT (Devlin et al., 2019) and Sentence-BERT (Reimers and Gurevych).



(a) D, E, K → B

(b) B, D, K → E

Figure 6: Comparison the performance of our model with and without self-training, accuracy w.r.t. training steps.

model becomes more elastic for fitting various distributions. The corresponding accuracy on the challenging adaptation tasks from Amazon-review to Skytrax-review increases by 0.6 percentage points on average (78.3% vs. 78.9%).

4.4. Modal Analysis

Hyperparameter analysis: In this section, we explore the sensitivity of our framework to the hyperparameter λ_d in Eq.4, which trade-offs the effect of feature structure matching loss (ℓ_{fsm}). We observe that ℓ_{fsm} is quite robust, and we select λ_d from $\{10^0, 10^1, 10^2, 10^3\}$. The adaptation results on Amazon review benchmark are reported in Figure 4. It can be noted that with the increasing of λ_d from 10^0 to 10^2, the corresponding performance on all the tasks are increased with a different extent, which indicates that ℓ_{fsm} is helpful to alleviate the domain shift and improve the transferability of the model. Besides, we further increase λ_d to a large value (*i.e.*, 10^3), the tendency of the accuracy curve is still stable, which verifies that the proposed ℓ_{fsm} is very robust.

We also validate the effectiveness of the number of fully-connected layers (N) within the dynamic parameters fusion module. As shown in Figure 5, we set N to $\{1, 4, 8, 16\}$ for comparisons. It is noted that $N = 1$ indicates that the whole model becomes static and can be regarded as the baseline. It is clear that increasing N can improve the performance on all the tasks, which implies that dynamically adjusting the network's parameters based on each input is more generalizable. In particular, in the task of adaptation from Amazon to Airport, the accuracy reaches 70.8% when $N = 8$. However, there is a decline when N goes larger, *i.e.*, $N = 16$. We consider that a large number of layers could increase the training difficulties. Consequently, we set $N = 4$ in all experiments. This analysis also shows that proper designation of the model and selection of hyperparameters can further improve the final results.

Orthogonal with self-training: Recently, self-training with pseudo-labels (Sohn et al., 2020) has been widely adopted in various semi-supervised learning and domain adaptation tasks. We show that our proposed framework is orthogonal with the self-training method. Specifically, we use model to infer pseudo-labels of the target data which will join the training during the adaptation. As shown in Figure 6, the accuracy of our model can be further improved with the help of self-training on both tasks.

Complementary to Sentence-BERT: Sentence-

BERT (Reimers and Gurevych) is a more advanced pretrained language model which is trained with siamese networks, and the extracted features should be more informative. As shown in Table 5, we investigate the performance by adopting the Sentence-BERT as the backbone, and observe around 2.6 percent points improvement on average on the Amazon benchmark.

5. Conclusion

In this work, we propose a novel framework for unsupervised multi-source sentiment adaptation. In contrast to traditional domain matching methods which may compromise performance in multi-source scenarios, we propose a feature structure matching constraint for more robust and generalizable adaptation. Besides, to achieve efficient adaptive tuning with the large pretrained language model, we propose a dynamic parameters fusion module to fully exploit the global information and adjust the model's parameters to fit various input. Experiments on multiple sentiment adaptation benchmarks and the ablation studies verify the effectiveness of our framework.

Acknowledgments. This work was supported in part by National Natural Science Foundation of China (Project No.62106136), in part by the Natural Science Foundation of Guangdong Province (Project No.2022A1515010434), in part by Shantou University under Project NTF20007 and Project NTF22012.

6. References

Erik Cambria, Yang Li, Frank Z. Xing, Soujanya Poria, and Kenneth Kwok. 2020. Senticnet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis. In *CIKM*, pages 105–114. ACM.

Chao Chen, Zhihang Fu, Zhihong Chen, Sheng Jin, Zhaowei Cheng, Xinyu Jin, and Xian-Sheng Hua. 2020a. Homm: Higher-order moment matching for unsupervised domain adaptation. In *AAAI*, pages 3422–3429.

Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. 2020b. Dynamic convolution: Attention over convolution kernels. In *CVPR*, pages 11027–11036.

Yong Dai, Jian Liu, Xiancong Ren, and Zenglin Xu. 2020. Adversarial training based multi-source unsupervised domain adaptation for sentiment analysis. In *AAAI*, pages 7618–7625.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics.

Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi, and Jianxin Liao. 2020. Adversarial and domain-aware BERT for cross-domain sentiment analysis. In *ACL*, pages 4019–4028.

Yanping Fu and Yun Liu. 2022. Contrastive transformer based domain adaptation for multi-source cross-domain sentiment classification. *Knowl. Based Syst.*, 245:108649.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. 2016. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NeurIPS*, pages 2672–2680.

Ishaan Gulrajani and David Lopez-Paz. 2021. In search of lost domain generalization. In *ICLR*.

Jiang Guo, Darsh J. Shah, and Regina Barzilay. 2018. Multi-source domain adaptation with mixture of experts. In *EMNLP*, pages 4694–4703.

Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. 2022. Dynamic neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(11):7436–7456.

Judy Hoffman, Mehryar Mohri, and Ningshan Zhang. 2018. Algorithms and theory for multiple-source adaptation. In *NeurIPS*, pages 8256–8266.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *ICML*, volume 97, pages 2790–2799.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, volume 9906, pages 694–711.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Rui Li, Cheng Liu, and Dazhi Jiang. 2022. Asymmetric mutual learning for multi-source unsupervised sentiment adaptation with dynamic feature network. In *COLING*, pages 6934–6943.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL/IJCNLP*, pages 4582–4597.

Yunsheng Li, Lu Yuan, Yinpeng Chen, Pei Wang, and Nuno Vasconcelos. 2021. Dynamic transfer for multi-source domain adaptation. In *CVPR*, pages 10998–11007.

Pingping Lin and Xudong Luo. 2020. A survey of sentiment analysis based on machine learning. In *NLPCC*, volume 12430, pages 372–387.

Hong Liu, Jianmin Wang, and Mingsheng Long. 2021. Cycle self-training for domain adaptation. In *NeurIPS*, pages 22968–22981.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *ACL*, pages 61–68.

Mingsheng Long, Yue Cao, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. 2019. Transferable representation learning with deep adaptation networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(12):3071–3085.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? In *Workshop BlackboxNLP@EMNLP*, pages 33–44.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, pages 3980–3990.

Maryem Rhanoui, Mounia Mikram, Siham Yousfi, and Soukaina Barzali. 2019. A cnn-bilstm model for document-level sentiment analysis. *Mach. Learn. Knowl. Extr.*, 1(3):832–847.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*.

Baochen Sun and Kate Saenko. 2016. Deep CORAL: correlation alignment for deep domain adaptation. In *ECCV Workshops*, volume 9915, pages 443–450.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune BERT for text classification? In *CCL*, volume 11856, pages 194–206. Springer.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 2022. LST: ladder side-tuning for parameter and memory efficient transfer learning. In *NeurIPS*.

Yosephine Susanto, Erik Cambria, Ng Bee Chin, and Amir Hussain. 2022. Ten years of sentic computing. *Cogn. Comput.*, 14(1):5–23.

Geng Tu, Jintao Wen, Cheng Liu, Dazhi Jiang, and Erik Cambria. 2022. Context- and sentiment-aware networks for emotion recognition in conversation. *IEEE Trans. Artif. Intell.*, 3(5):699–708.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*, pages 5998–6008.

Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. 2022. A survey on sentiment analysis methods, applications, and challenges. *Artif. Intell. Rev.*, 55(7):5731–5780.

Garrett Wilson and Diane J. Cook. 2020. A survey of unsupervised deep domain adaptation. *ACM Trans. Intell. Syst. Technol.*, 11(5):51:1–51:46.

Canwen Xu and Julian J. McAuley. 2023. A survey on dynamic neural networks for natural language processing. In *Findings EACL*, pages 2325–2336.

Brandon Yang, Gabriel Bender, Quoc V. Le, and Jiquan Ngiam. 2019a. Condconv: Conditionally parameterized convolutions for efficient inference. In *NeurIPS*, pages 1305–1316.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, pages 5754–5764.

Han Zhao, Shanghang Zhang, Guanhang Wu, José M. F. Moura, João Paulo Costeira, and Geoffrey J. Gordon. 2018. Adversarial multiple source domain adaptation. In *NeurIPS*, pages 8568–8579.

Sicheng Zhao, Xiangyu Yue, Shanghang Zhang, Bo Li, Han Zhao, Bichen Wu, Ravi Krishna, Joseph E. Gonzalez, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia, and Kurt Keutzer. 2022. A review of single-source deep unsupervised visual domain adaptation. *IEEE Trans. Neural Networks Learn. Syst.*, 33(2):473–493.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. 2021. Domain adaptive ensemble learning. *IEEE Trans. Image Process.*, 30:8008–8018.

Yftah Ziser and Roi Reichart. 2018. Pivot based language modeling for improved neural domain adaptation. In *NAACL-HLT*, pages 1241–1251.